(12) UK Patent Application (19) GB (11) 2 315 638 (13) A

(43) Date of A Publication 04.02.1998

(21) Application No 9615153.5

(22) Date of Filing 19.07.1996

(71) Applicant(s)
Telefonaktiebolaget L M Ericsson

(Incorporated in Sweden)

S-126 25, Stockholm, Sweden

(72) Inventor(s)
Bart Jellema
Rene Peeren
Louise Croughan
Freek Aben

(74) Agent and/or Address for Service
Haseltine Lake & Co
Imperial House, 15-19 Kingsway, LONDON, WC2B 6UD, United Kingdom

(51) INT CL⁶
H04L 12/08 1/00 , H04Q 3/00

(52) UK CL (Edition P)
H4K KF42
H4P PPEC

(56) Documents Cited
EP 0570220 A2 EP 0418866 A2 EP 0358293 A2
WO 95/06384 A1 WO 90/06647 A1 US 5425025 A

(58) Field of Search
UK CL (Edition O) **H4K KF42 KOD4 KOD8 , H4P PENL PPEC**
INT CL⁶ **H04L 1/00 1/16 1/18 12/08 , H04M 11/06 , H04Q 3/00**
ONLINE : WPI

(54) **Communications system with sequential message numbering; Intelligent Networks**

(57) In a duplex communications system with asynchronous exchange of messages between two stations, such as between a service switching point SSP and a service control point SCP in an Intelligent Network, each message sent from a station includes a sequentially assigned number indicated by a sending ID parameter SID and also includes a receiving parameter LPID indicating the number of the most recent message received from the other station. For example, the SCP sends a message with SID= 1*,LPID = 1 in response to message number 1 from the SSP. If the SSP sends a message SID=3,LPID=1* before receiving a message SID=2*,LPID=2 sent by the SCP in response to the previous message (SID=2,LPID=1*) sent by the SSP, the SSP can detect such message "crossing" (ie. the stations are effectively out of synchronisation) because the LPID =2 parameter indicates a response to message SID =2, rather than message SID =3, from the SSP. The SSP may then ignore message SID=2*,LPID=2 from the SCP and await the SCP's response SID=3*,LPID=3 to the SSP's most recent message (SID =3,LPID = 1*). Because of the sequential numbering, it can also be detected when a message has been lost during transmission (Fig.6).

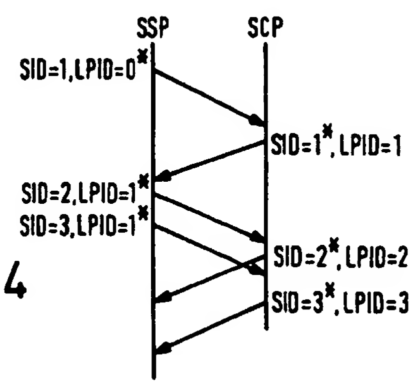


FIG. 4

GB 2 315 638 A

FIG. 1
(PRIOR ART)

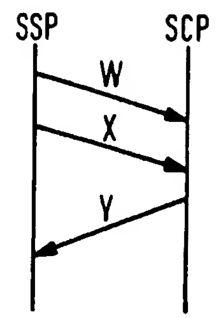


FIG. 2
(PRIOR ART)

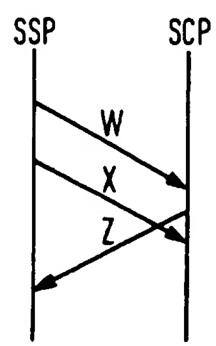
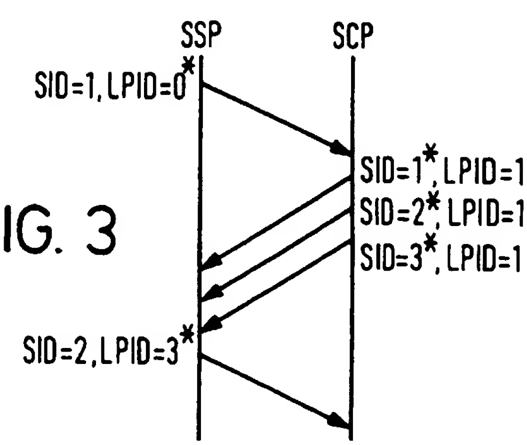


FIG. 3



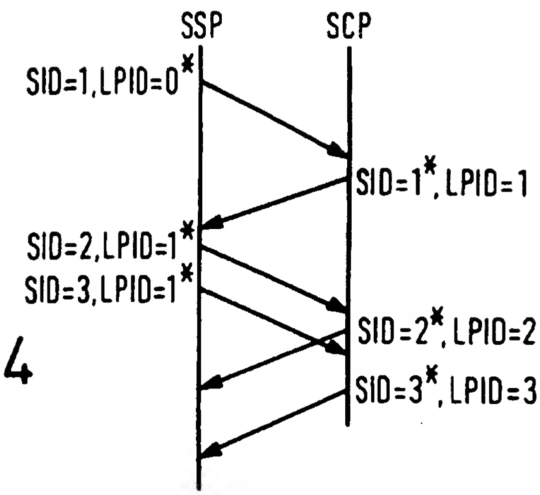


FIG. 4

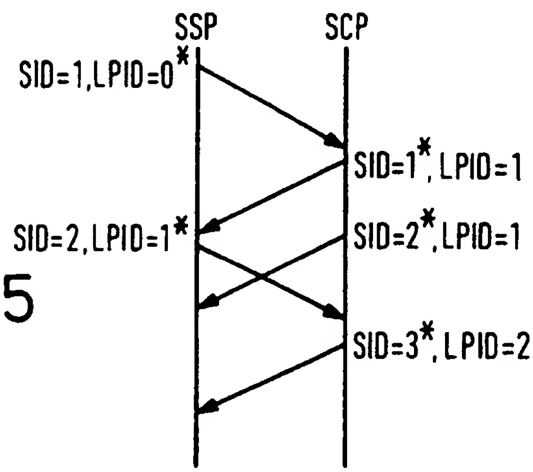


FIG. 5

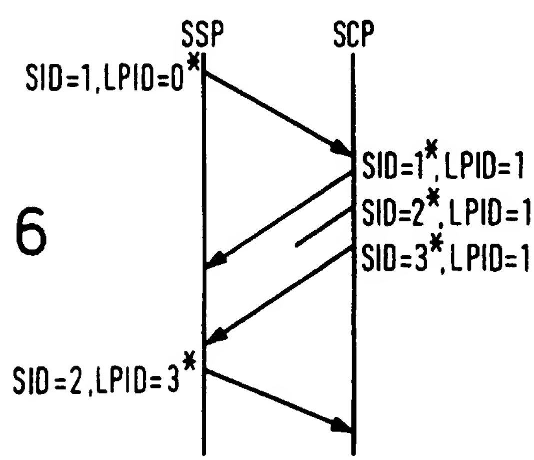


FIG. 6

2315638

SYNCHRONISATION CHECKING

This invention relates to a mechanism for checking synchronisation between applications in a network architecture. In particular, the invention relates to a mechanism for checking synchronisation between messages sent between a Service Control Point and a Service Switching Point in an Intelligent Networks architecture.

An Intelligent Networks architecture includes a Service Control Point (SCP), which typically has a large number of Service Switching Points (SSPs) connected to it. Each SSP is a switching system that can intercept telephone calls, and query the SCP. The SCP contains service specific logic and data, but allows it to return instructions to the SSP on how to deal with the intercepted call.

The link between the SSP and SCP uses a full duplex communication protocol, meaning that each entity can be sending a message to the other at the same time. This can result in synchronisation problems, because, when an entity receives a message, it is not apparent whether that message is in response to a recently sent message, or whether it was sent before its own most recently sent message was received at the other entity. This problem is illustrated with reference to Figures 1 and 2 of the accompanying drawings. In the example shown in Figure 1, the SSP sends two messages W and X to the SCP in quick succession. For example, the SSP might be reporting events such as an answer by the called party, quickly followed by disconnection by the calling party. In Figure 1, the SCP sends message Y in response to report X. Thus, the system is synchronised.

However, in Figure 2, where the SSP sends the same two messages W and X, the SCP sends message Z in response to report W. By the time message Z is

received by the SSP, it is inappropriate, because the further event reported in message X has already occurred. However, the SSP has no way of knowing that the system is out of synchronisation.

In accordance with the present invention, the communications protocol between the two entities provides that the messages carry synchronisation information. In particular, each message indicates the most recently received message from the other party, to which it is responding. This has the advantage that it is possible to maintain synchronisation, preventing the SSP or SCP from acting on unreliable messages, improving the operation of the network.

For a better understanding of the present invention, reference will now be made, by way of example, to the accompanying drawings, in which:

Figure 1 is a schematic illustration of messages sent in a prior art protocol;

Figure 2 is a schematic diagram of a different set of messages sent with the prior art protocol;

Figure 3 is a first illustrative example of messages being sent in accordance with the invention;

Figure 4 is a second illustrative example of messages being sent in accordance with the invention;

Figure 5 is a third illustrative example of messages being sent in accordance with the invention; and

Figure 6 is a fourth illustrative example of messages being sent in accordance with the invention.

Figures 3-6 are illustrative examples of messages being exchanged between a SSP and SCP. In these diagrams, the SSP is represented on the left, and the SCP is represented on the right. In addition, time is represented vertically, with later times appearing lower on the diagrams. Thus, a message from SSP to SCP is represented by a line diagonally downwards and to

the right, with the size of the downwards component representing the time taken for the message to go across the network from SSP to SCP.

In accordance with the invention, each message sent over the SSP-SCP interface has a number assigned to it by the sending entity. In Figures 3-6, these numbers are indicated by the parameter SID (Sending ID). The numbers are assigned sequentially, and transmitted with the messages. In Figures 3-6, the Sending ID's of messages from SSP to SCP are indicated by plain numerals, while Sending ID's assigned to messages from SCP to SSP are indicated by numerals with asterisks (e.g. 1*, etc).

Each message transferred between the SSP and SCP also contains a parameter LPID (Last Processed ID), which indicates the Sending ID of the last message which was received from the receiving entity and processed by the sending entity.

Thus, when a receiving entity detects that the LPID parameter, in a message which it has just received, does not correspond with the SID parameter in the last message which it sent, it knows that there is a lack of synchronisation. This allows the receiving entity to decide not to process the received message, until another message is received from the sending entity, indicating that its own last sent message has been processed.

Thus, in the example illustrated in Figure 3, a query is sent first from the SSP to the SCP, with the SID parameter = 1, and the LPID parameter = 0*. In response, the SCP sends three instructions, identified respectively with the parameter values SID = 1*, LPID = 1 for the first instruction; SID = 2*, LPID = 1 for the second instruction; and SID = 3*, LPID = 1 for the third instruction. Thus, when the SSP receives these three instructions, it is able to determine, firstly,

that it has received the three instructions in the sequence, and that those instructions were correctly sent by the SCP in response to the SSP's own most recently sent message, identified by SID = 1. When reporting the next event, the SSP therefore sends a message with parameters SID = 2, LPID = 3*, indicating that the third instruction sent from the SCP, having parameters SID = 3*, LPID = 1 was the last message it received from the SCP.

In the example shown in Figure 4, there is a loss of synchronisation, caused by messages crossing in the network.

Thus, the first event is reported from the SSP to the SCP in a message with parameters SID = 1, LPID = 0*. The SCP responds with an instruction with parameters SID = 1*, LPID = 1. The SSP then reports two events to the SCP, in respective messages with parameters SID = 2, LPID = 1* and SID = 3, LPID = 1*. However, the SCP responds to the earlier of these two messages, before it has processed the later, and therefore it replies with a message with parameters SID = 2*, LPID = 2, this latter parameter indicating that the message from the SSP with the Sending ID = 2 was the last message which it processed. In this case, by contrast with the prior art, the SSP is able to detect that synchronisation has been lost, in that the SCP has sent instructions, without having processed the message most recently sent from the SSP to the SCP. The SSP therefore waits for further instructions before acting on the instructions contained in the message from the SCP. In due course, the SCP sends a further message, with parameters SID = 3*, LPID = 3, this latter parameter indicating that the message from the SSP with the Sending ID = 3 has now been processed. The SSP is then able to detect that the SSP and SCP are back in synchronisation, and is able to execute the SCP

instructions.

Thus, the system in accordance with the invention is able to maintain synchronisation, even when messages are sent with small time intervals therebetween.

Figure 5 illustrates another example of the use of the invention. Again, an event is first reported from the SSP to SCP, in a message with parameters SID = 1, LPID = 0*. In response, the SCP sends instructions, in a message with parameters SID = 1*, LPID = 1. However, in this case, the SSP reports that there is an error in the message received from the SCP, and sends a further message with parameters SID = 2, LPID = 1*. Meanwhile, before receiving this error message from the SSP, the SCP sends a second instruction, in a message with parameters SID = 2*, LPID = 1. In this case, when that message is received by the SSP, the SSP is able to detect that its most recent message has not been processed by the SCP, and so is able to wait until it receives a message indicating that its own most recently sent message has been processed, i.e. with a parameter LPID = 2. In due course, the SCP acts on the error message sent from the SSP, and sends a new message with parameters SID = 3*, LPID = 2. The SSP is then able to detect that the system is back in synchronisation, and executes the instructions sent from the SCP. Again, the system is able to allow synchronisation to be regained.

Figure 6 illustrates another use of the invention, where a message is lost during transmission. Again, an event is first reported from the SSP to the SCP, in a message with parameters SID = 1, LPID = 0*. In response, the SCP sends three instructions to the SSP, in messages with parameters SID = 1*, LPID = 1 for the first message; SID = 2*, LPID = 1 for the second message; and SID = 3*, LPID = 1 for the third message. However, only the first and third messages are received

by the SSP, with the second message somehow being lost in the network. In this case, the SSP is able to detect that a message is missing, because the SID parameters in the messages which it has received are not sequential. The SSP is therefore able to report an error to the SCP, which it does in a message with parameters SID = 2, LPID = 3*.

Therefore, there is disclosed a system in which synchronisation errors are able to be detected and, after detection, are able to be corrected.

Although the invention has been described herein with reference to an Intelligent Networks architecture, it will be appreciated that a similar protocol may be used in any duplex communications system relying on an asychronous exchange of messages.

CLAIMS

1. A method of operating a duplex communications system between a first party and a second party, in which messages sent from the first party to the second party include a sequential sending parameter, and messages sent from the second party to the first party include a receiving parameter corresponding to the sending parameter of the message most recently received by the second party from the first party.

2. A method as claimed in claim 1, wherein messages sent from the second party to the first party also include a sending parameter, and messages sent from the first party to the second party also include a receiving parameter corresponding to the sending parameter of the message most recently received by the first party from the second party.

3. A method as claimed in claim 1 or claim 2, wherein the first party is a switching point in a telecommunications network, and the second party is a control point.

4. A method as claimed in claim 1, wherein, when the first party receives a first message from the second party, in which the receiving parameter does not correspond to the sending parameter of the message most recently sent from the first party to the second party, the first party delays acting on the first message.

5. A method as claimed in claim 4, wherein the first party awaits a second message, in which the receiving parameter does correspond to the sending parameter of the message most recently sent from the first party to the second party.

6. A method as claimed in claim 1, wherein, when the second party detects that it has received no message from the first party with a sending parameter in an expected sequence, the second party sends an error message to the first party.

The Patent Office

8

Application No:	GB 9615153.5	**Examiner:**	M J Billing
Claims searched:	1 to 6	**Date of search:**	30 October 1996

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): H4K KF42, KOD4, KOD8; H4P PENL, PPEC.

Int Cl (Ed.6): H04L 1/00, 1/16, 1/18, 12/08; H04M 11/06; H04Q 3/00.

Other: ONLINE : WPI.

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	EP0570220A2	(IBM) - column 3 lines 38-41	1 at least
X	EP0418866A2	(NIPPON TELEGRAPH) - column 5 line 29 to column 6 line 17	1,2,6 at least
X	EP0358293A2	(DIGITAL) - column 5 lines 6-56, column 20 line 20 to column 21 line 16	1,4,5 at least
X	WO95/06384A1	(TRANSACTION TECHNOLOGY) - page 14 line 1 to page 15 line 13	1,2,6 at least
X	WO90/06647A1	(COMPUQUEST) - page 8 lines 20-25, page 12 lines 1-6	1,2 at least
X	US5425025	(FUJITSU) - Figs.10,11	1,2,6 at least

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.
& Member of the same patent family
A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.